DOCUMENT RESUME

ED 093 336 IR 000 823

AUTHOR Tritchler, David L.; Pedrini, D. T.
TITLE Computer Technology, Large Samples, and Fisher's Exact Probability Test.
PUB DATE 1 Aug 74
NOTE 19p.

EDRS PRICE MF-$0.75 HC-$1.50 PLUS POSTAGE
DESCRIPTORS *Computer Programs; *Hypothesis Testing; *Probability; Program Descriptions; *Statistical Analysis
IDENTIFIERS *Fisher Exact Probability Test

ABSTRACT

The Fisher Exact Probability Test considers independent groups and dichotomous criterion in a 2 by 2 (fourfold) contingency table. Fisher's test requires a determination of the combined probability of the observed frequency distribution and of the more extreme (in the same direction) frequency distributions. Typically, Fisher's test is for a small sample, with a total size of 30 or less. Otherwise, the computations required are much too tedious, and chi-square is used. This paper presents a mathematical discussion as well as a computer program for Fisher's test. Included are usage instructions and an illustrative problem. Depending on the frequency pattern, this program can handle sample sizes of about 500. Therefore, in most instances, a researcher does not have to resort to approximation methods. (Author)

ED 093336

COMPUTER TECHNOLOGY, LARGE SAMPLES, AND FISHER'S EXACT PROBABILITY TEST

BEST COPY AVAILABLE

David L. Tritchler and D. T. Pedrini

University of Nebraska at Omaha, Omaha, Nebraska 68101

Abstract

The Fisher Exact Probability Test considers independent groups and dichotomous criterion in a 2 by 2 (fourfold) contingency table. Fisher's Test requires a determination of the combined probability of the observed frequency distribution and of the more extreme (in the same direction) frequency distributions. Typically, Fisher's Test is for a small sample, with a total size of 30 or less. Otherwise, the computations required are much too tedious and chi square is used. This paper presents a mathematical discussion as well as a computer program for Fisher's Test. Included are usage instructions and an illustrative problem. Depending upon the frequency pattern. this program can handle sample sizes of about 500. Therefore, in most instances, a researcher does not have to resort to approximation methods.

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

BEST COPY AVAILABLE

TR 000 893

CONTENTS

1.	FUNCTION	3
2.	MATHEMATICAL DISCUSSION	4
3.	SUMMARY OF CALCULATION PROCEDURE	7
4.	USAGE INSTRUCTIONS	8
5.	SAMPLE PROBLEM	9
6.	FLOWCHART	10
7.	PROGRAM LISTING	12
8.	SUBROUTINES	17
9.	MEMORY REQUIREMENTS AND RUN TIME	18
10.	BIBLIOGRAPHY	19

1. FUNCTION

A common statistical problem is the determination of the significance of a difference between 2 independent samples or groups for which the scores fall into 2 mutually exclusive classes. This problem is commonly depicted in the form of a fourfold or a 2X2 contingency table. Fisher's Test is a technique for analyzing such data, and provides an exact probability. This test becomes impractical for hand computation in the case of large sample size, and usually an approximate method such as chi square is opted for.

This paper presents a computer program for Fisher's Exact Probability Test (FISHEPT) which can handle sample sizes of up to about 500. Thus in most instances the researcher need not turn to approximate methods.

2. MATHEMATICAL DISCUSSION

Fisher's Exact Test determines whether 2 groups differ in the proportion in which they fall into 2 mutually exclusive classes. If the marginal totals of a given 2X2 contingency table are considered fixed, the exact probability of a given set of frequencies, given the null hypothesis of no difference in population proportions, is given by the hypergeometric distribution.

Let + and - be the 2 mutually exclusive outcomes. Then a given result is of the form:

	+	-	
Group I	Cell A a	Cell B b	a+b
Group II	Cell C c	Cell D d	c+d
	a+c	b+d	

where a, b, c, and d are cell frequencies. The probablity of a given set of observed frequencies is given by

$$p = \frac{(a+b)!\,(c+d)!\,(a+c)!\,(b+d)!}{N!\,a!\,b!\,c!\,d!} \qquad (A)$$

This formula gives the probability of the observed frequencies. However, the null hypothesis requires a determination of the combined prcbability of the observed set of frequencies and all sets of frequencies more extreme than that observed. For example, suppose the following 2X2 contingency table was observed:

	+	-	
Group I	2	15	17
Group II	10	3	13
	12	18	

Since the marginal totals remain fixed (i.e., 17, 13, 12, 18), there are 2 outcomes more extreme than the observed outcome:

	+	-	
Group I	1	16	17
Group II	11	2	13
	12	18	

and

	+	-	
Group I	0	17	17
Group II	12	1	13
	12	18	

The sum of the probabilities of the 3 outcomes is the probability of observing an outcome as extreme as, or more extreme than (i.e., extreme in the same direction) the observed outcome.

In order to discuss the computation in a general sense and to formalize the concept of "more extreme outcome," let us fix a standard arrangement for a 2X2 contingency table. The standard format will be:

	+	-	
Group I	a Cell A	b Cell B	a+b
Group II	c Cell C	d Cell D	c+d
	a+c	b+d	

where a, b, c, d are the cell frequencies and ad $\leq$ bc

If ad = bc, then a is the smallest of a, b, c, and d.
If ad $<$ bc, then a $\leq$ d.

when our data are arranged in a 2X2 table conforming to the above rules, all more extreme results are abtained by successively decrementing the frequency of cell A by 1 until a=0. For each value of a, the values b, c, and d are fixed by the marginal totals. This convention permits unambiguous discussion and is also useful for identifying our input data to the computer program. The numerical example of a 2X2 table, given on page nine, conforms to our convention.

Let a = M in a 2X2 table of observed data. Then we need the sum of the probabilities of M + 1 tables (i.e., all possibilities from a = 0 to a = M). To avoid confusion, let us denote these tables as $T_0, T_1, \ldots, T_M$ Where T_K is the table for which a = K. Thus T_M is the observed occurance. a_M will be the frequency in cell A of T_M and likewise for b_M, c_M, and d_M.

Let P_K be the probability of observing T_K, given the null hypothesis. Then the probability of an occurance as extreme as, or more extreme than, the observed occurrance is given by:

$$P = \sum_{K=0}^{M} P_i$$ where M is the frequency in cell A for the observed data.

Each P_K could be formed by using formula A for each T_K, but a more efficient method exists.

For P_0, the fact that $a_0 = 0$ may be used to simplify formula A. Thus, for T_0:

$$P_0 = \frac{(a_0 + b_0)!\,(c_0 + d_0)!\,(a_0 + c_0)!\,(b_0 + d_0)!}{N!\,a_0!\,b_0!\,c_0!\,d_0!}$$

$$= \frac{b_0!\,(c_0 + d_0)!\,c_0!\,(b_0 + d_0)!}{N!\,0!\,b_0!\,c_0!\,d_0!}$$

$$= \frac{(c_0 + d_0)!\,(b_0 + d_0)!}{N!\,d_0!} \qquad (B)$$

Next we derive an inductive formula for P_{k+1} in terms of P_k. The terms $(a_k + b_k)$, $(c_k + d_k)$, $(a_k + c_k)$, and $(b_k + d_k)$ are fixed (i.e., the same for all k), since they are marginal totals. Therefore let us label these $(a + b)$, $(c + d)$ etc. for all k. For P_{k+1}, Formula A becomes:

$$P_{k+1} = \frac{(a + b)!\,(c + d)!\,(a + c)!\,(b + d)!}{N!\,(a_k + 1)!\,(b_k - 1)!\,(c_k - 1)!\,(d_k + 1)!}$$

because when we add 1 to the frequency of cell A we are forced by the marginal totals to add 1 to cell D and subtract 1 from cells B and C.

Dividing we see that:

$$\frac{P_{k+1}}{P_k} = \frac{A_k!\,b_k!\,c_k!\,d_k!}{(a_k+1)!\,(b_k-1)!\,(c_k-1)!\,(d_k+1)!}$$

Therefore:

$$\frac{P_{k+1}}{P_k} = \frac{b_k c_k}{(a_k+1)(d_k+1)}$$

And,

$$P_{k+1} = \frac{b_k c_k}{(a_{k+1})(d_{k+1})} P_k \qquad (C)$$

Thus, once we have determined P_0 by formula B, we can, by means of formula C, inductively determine P_1, then P_2, and so on until P_M is found. Then

$$P = \sum_{k=0}^{M} P_k$$

3. SUMMARY OF CALCULATION PROCEDURE

A form of formula B (see Mathmetical Discussion) was used to determine P_0. The derivation is as follows:

$$P_0 = \frac{(c_0+d_0)!\,(b_0+d_0)!}{(a_0+b_0+c_0+d_0)!\;d!}$$

$$= \frac{(c_0+d_0)!}{d_0!} \cdot \frac{(b_0+d_0)!}{(a_0+c_0+b_0+d_0)!}$$

$$= \left[\prod_{k=0}^{c_0-1} c_0+d_0-k\right] \div \left[\prod_{j=0}^{c_0-1} a_0+b_0+c_0+d_0-j\right]$$

Formula C is then applied iteratively to find P. This algorithm is computationally efficient and turns out to provide accuracy. If a and b are the frequencies in the observed cells A and B, and n is the number of significant digits of a double precision number for the machine being used, the result will have a maximum relative error of

$$\frac{4a + 2c}{10^{n-1}} .$$

Due to the large values obtained in working with factorials, exponent overflow is a problem. To cope with this, values are repeatedly scaled to prevent exponent overflow. This eventually breaks down, dependent on the sample size and the distribution among cells. However, problems with N = 500 have been analyzed on the IBM 360/65.

4. USAGE INSTRUCTIONS

On one data card, enter the frequencies for one group and then enter the frequencies for the other group in the same order, i.e. if a,b,c,d are entered, a and c should be frequencies for one class of scores and b and d should be frequencies for the other class of scores, where a and b are frequencies of the same group and c and d are the frequencies of the other group.

The program automatically puts the problem in a standard form (see Mathematical Discussion). The printed output from the program will include the depiction of the problem in standard form.

The 4 values may be punched on the data card in any card columns, as long as one or more blanks separate them and they are in the order described above. No decimal points or commas are allowed, only integer values. A legal data card will consist of 4 integer values with blanks separating them, and no other characters. Multiple cards may be submitted, with each card corresponding to one Fisher's Exact Test to be performed.

For example, if an outcome was:

	+	-
GROUP I	3	4
GROUP II	2	9

any of the following four sets of data will describe the outcome to the program.

1) 3 4 2 9

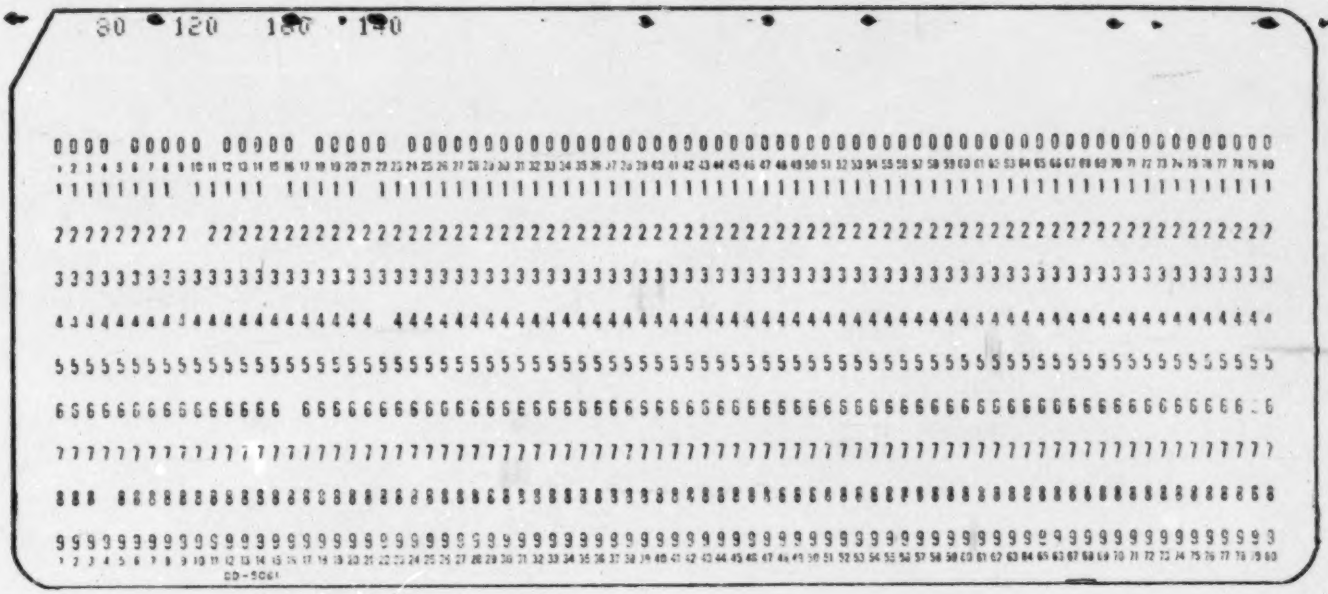
2) 4 3 9 2

3) 2 9 3 4

4) 9 2 4 3

5. SAMPLE PROBLEM

(a) Input



(b) Output

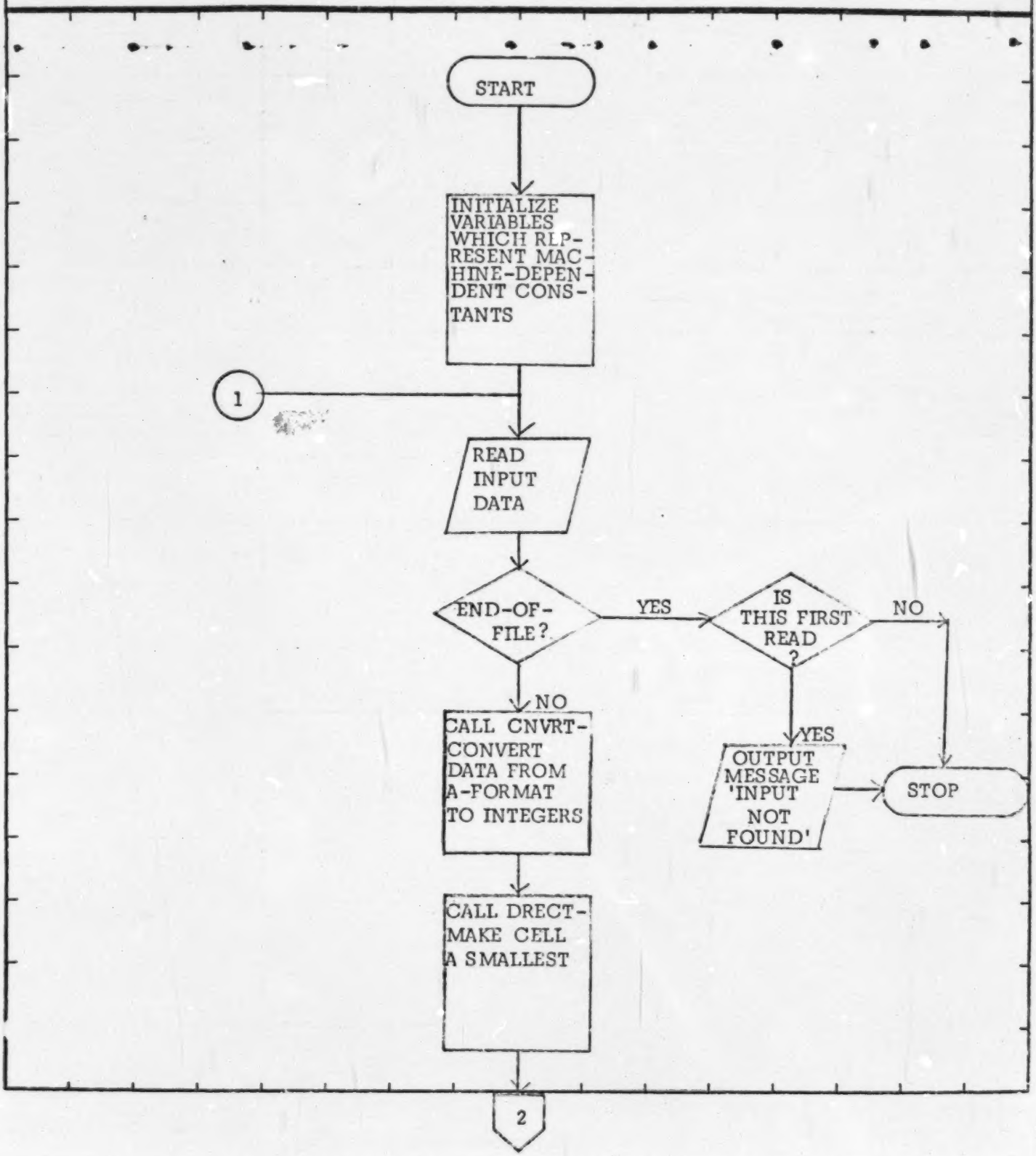
OUTCOME

		+	-	
G	X	80	120	200
R				
O				
U				
P	Y	160	140	300
		240	260	500

P (OCCURANCE OF A DIFFERENCE THIS EXTREME OR GREATER) = 0.0023
NOTE- THIS IS A ONE-TAILED TEST

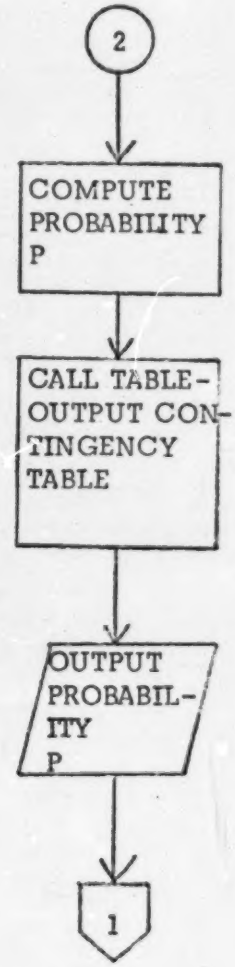
6. Flowchart: (Prepared by Vince Scarpello, University of Nebraska Medical Center

Project Leader Dave Tritchler System/Job/Program Name FISHET
Prepared By Vince Scarpello Date 10/04/73 Page 1 of 2



Project Leader Dave Tritchler System/Job/Program Name FISHET.

Prepared By Vince Scarpello Date 10/04/73 Page 2 of 2



```
C-----------------------------------------------------------------------FET00001
C      AUTHOR: DAVID L. TRITCHLER                                       FET00002
C              UNIVERSITY OF NEBRASKA COMPUTER NETWORK                  FET00003
C              OMAHA COMPUTING FACILITY                                 FET00004
C      PURPOSE: PERFORM FISHER'S EXACT TEST ON A 2-BY-2 CONTINGENCY TABLEFET00005
C-----------------------------------------------------------------------FET00006
C                                                                       FET00007
C                                                                       FET00008
C                                                                       FET00009
C         THE FOLLOWING CARD SHOULD BE USED ONLY IN IBM 360/370 OS      FET00010
C      FORTRAN IV                                                       FET00011
       EXTERNAL USERR                                                   FET00012
C                                                                       FET00013
C                                                                       FET00014
C                                                                       FET00015
       DOUBLE PRECISION P,PK,DOVFL,DEXP,DJA,DJB,DJC,DJD,D70,D0          FET00016
C-----------------------------------------------------------------------FET00017
C      IF YOU DO NOT DESIRE DOUBLE PRECISION THEN REMOVE THE ABOVE CARD FET00018
C      NOTE THAT ALL VARIABLES WHOSE NAMES BEGIN WITH I, J, K, L, M, OR NFET00019
C      ARE INTEGERS. ALL OTHERS ARE DOUBLE PRECISION REALS.             FET00020
C-----------------------------------------------------------------------FET00021
       COMMON IA,IC,IAB,ICD,ICO,IOUT,N80,NA,NC,NAB,NCD,NAC              FET00022
       COMMON KARD(200)                                                 FET00023
       COMMON IE                                                        FET00024
       DATA IDLLR/1H$/                                                  FET00025
C                                                                       FET00026
C                                                                       FET00027
C                                                                       FET00028
C         THE FOLLOWING CARD SHOULD BE USED ONLY IN IBM 360/370 OS       FET00029
C      FORTRAN IV                                                       FET00030
       CALL ERRSET(208,255,-1,1,USERP)                                  FET00031
C                                                                       FET00032
C                                                                       FET00033
C                                                                       FET00034
C-----------------------------------------------------------------------FET00035
C      NOTE THAT THE MINIMUM RECORD SIZE FOR OUTPUT IS 67 CHARACTERS    FET00036
C      BECAUSE OF THE OUTPUT FORMATS IN THIS PROGRAM & SUBROUTINE TABLE FET00037
C      THE FIRST CHARACTER IS ALWAYS FOR CARRIAGE CONTROL               FET00038
C-----------------------------------------------------------------------FET00039
   22 FORMAT ( 45HOP (OCCURENCE OF A DIFFERENCE THIS EXTREME OR,        FET00040
     C 12H GREATER ) =,F9.4,/,3X,32HNOTE - THIS IS A ONE-TAILED TEST)   FET00041
  111 FORMAT (16H INPUT NOT FOUND  )                                    FET00042
C-----------------------------------------------------------------------FET00043
C      IN ORDER TO CHANGE THE INPUT RECORD SIZE                         FET00044
C      MERELY CHANGE  THE NEXT TWO CARDS:                               FET00045
  211 FORMAT (80A1)                                                     FET00046
       N80 = 80                                                         FET00047
C      EXAMPLE: IF THE MAXIMUM SIZE RECORD ALLOWED ON THE INPUT MEDIUM ISFET00048
C      72 CHARACTERS, SUBSTITUTE THE FOLLOWING TWO STATEMENTS ...       FET00049
C 211 FORMAT (72A1)                                                     FET00050
C      N80 = 72                                                         FET00051
C      ... DO NOT CHANGE THE NAME OF THE VARIABLE N80                   FET00052
C-----------------------------------------------------------------------FET00053
       NUMIN = 0                                                        FET00054
C-----------------------------------------------------------------------FET00055
C      IN = INPUT LOGICAL UNIT NUMBER = 5                               FET00056
C      IOUT = OUTPUT LOGICAL UNIT NUMBER = 6                            FET00057
C-----------------------------------------------------------------------FET00058
       IN = 5                                                           FET00059
       IOUT = 6                                                         FET00060
C-----------------------------------------------------------------------FET00061
```

```
C     D70 = MAXIMUM MAGNITUDE OF A DOUBLE-PRECISION VARIABLE                FET00062
C     DOVFL = A CONSTANT USED TO MAKE SURE THAT FACTORIALS DO NOT           FET00063
C             BECOME TOO LARGE                                              FET00064
C     DO = THE CONSTANT 1.0 FOR DOUBLE-PRECISION ARITHMETIC                 FET00065
C     IF ANY VALUES BELOW EXCEED YOUR COMPUTER'S MAXIMUM MAGNITUDE OR       FET00066
C     IF YOU DO NOT DESIRE DOUBLE PRECISION THEN SUBSTITUTE                 FET00067
C     OTHER CONSTANTS IN THE NEXT 3 ARITHMETIC ASSIGNMENT STATEMENTS        FET00068
C-----------------------------------------------------------------------FET00069
      D70 = 1.D75                                                           FET00070
      DOVFL = 1.D10                                                         FET00071
      DO = 1.D0                                                             FET00072
C-----------------------------------------------------------------------FET00073
C     NOTE THAT IF YOUR FORTRAN DOES NOT HAVE AN END-OF-FILE                FET00074
C     INDICATOR, A RECORD WITH A '$' AS THE FIRST CHARACTER WILL ALSO       FET00075
C     TERMINATE EXECUTION                                                   FET00076
C-----------------------------------------------------------------------FET00077
    1 READ (IN,211,END=99) (KARD(I),I=1,N80)                                FET00078
      IE = 0                                                                FET00079
      IF (KARD(1) - IDLLR) 2,99,2                                           FET00080
    2 NUMIN = 1                                                             FET00081
      CALL CNVRT                                                            FET00082
      NA = IA                                                               FET00083
      NC = IC                                                               FET00084
      NAB = IAB                                                             FET00085
      NCD = ICD                                                             FET00086
      NAC = IA + IC                                                         FET00087
      CALL DRECT                                                            FET00088
C-----------------------------------------------------------------------FET00089
C     NOTE THAT NP1 = IAB + ICDP1 > ICDP1   BECAUSE IAB > 0                 FET00090
C-----------------------------------------------------------------------FET00091
      ICDP1 = ICD + 1                                                       FET00092
      NP1 = IAB + ICDP1                                                     FET00093
      P =      DO                                                           FET00094
      PK =     DO                                                           FET00095
      DEXP=  D70/DOVFL                                                      FET00096
      ICO = IA + IC                                                         FET00097
      DO 10 I = 1, ICO                                                      FET00098
      P = P * (ICDP1 - I )                                                  FET00099
C-----------------------------------------------------------------------FET00100
C     CHECK FOR OVERFLOW                                                    FET00101
C-----------------------------------------------------------------------FET00102
      IF(P -    DOVFL ) 4,3,3                                               FET00103
    3 P = P/DOVFL                                                           FET00104
      DEXP = DEXP * DOVFL                                                   FET00105
    4 PK = PK * ( NP1 - I )                                                 FET00106
C-----------------------------------------------------------------------FET00107
C     CHECK FOR OVERFLOW                                                    FET00108
C     NOTE THAT PK WILL OVERFLOW BEFORE P BECAUSE NP1 GREATER THAN ICDP1    FET00109
C-----------------------------------------------------------------------FET00110
      IF(PK -    DOVFL ) 10,8,8                                             FET00111
    8 PK = PK/DOVFL                                                         FET00112
      DEXP = DEXP/DOVFL                                                     FET00113
   10 CONTINUE                                                              FET00114
      P = P / PK                                                            FET00115
      P = P * DEXP                                                          FET00116
      PK = P                                                                FET00117
      DO 20 M = 1, IA                                                       FET00118
       I = M - 1                                                            FET00119
       DJA = I                                                              FET00120
       DJC = ICO - I                                                        FET00121
       DJD = ICD -DJC                                                       FET00122
       DJB = IAB -DJA                                                       FET00123
```

```
 10    CONTINUE                                                          FET00124
 20   INT(N) = 0                                                         FET00125
      DO 60 JJ = I,N80                                                   FET00126
       J = JJ                                                            FET00127
       KAR = KARD(J)                                                     FET00128
       DO 50 K = 1, 10                                                   FET00129
        IF ( KAR -     NUM(K) ) 50,30,50                                 FET00130
 30     INT(N) = INT(N) * 10 + K -1                                       FET00131
        GO TO 60                                                         FET00132
 50     CONTINUE                                                          FET00133
 55    GO TO 70                                                           FET00134
 60    CONTINUE                                                           FET00135
 70   ISTRT = J                                                           FET00136
 80   CONTINUE                                                            FET00137
     INT(4) = INT(4) + INT(3)                                            FET00138
     X = INT(3)                                                          FET00139
     INT(3) = INT(2) * INT(1)                                            FET00140
     INT(2) = X                                                          FET00141
     RETURN                                                              FET00142
     END                                                                 FET00143
     SUBROUTINE DRECT                                                    FET00144
C-------------------------------------------------------------------------FET00145
C      AUTHOR: DAVID L. TRITCHLER                                        FET00146
C      PURPOSE: MANIPULATE TABLE SO THAT CELL A IS THE SMALLEST          FET00147
C               CELL IN THE LESSER OF THE TWO DIAGONALS                  FET00148
C-------------------------------------------------------------------------FET00149
     COMMON IA, IC, IAB, ICD, ICO                                        FET00150
     COMMON /SWHCOM/ K(4)                                                FET00151
     K(1) = IA                                                           FET00152
     K(2) = IAB - IA                                                     FET00153
     K(3) = IC                                                           FET00154
     K(4) = ICD - IC                                                     FET00155
     IF (K(1) * K(4) - K(2) * K(3)) 10, 50, 30                           FET00156
 10  IF (K(1) - K(4)) 90, 90, 11                                         FET00157
 11  CALL  SWTCH(1,4,3,2)                                                FET00158
     GO TO 90                                                            FET00159
C                                                                        FET00160
C                                                                        FET00161
 30  CALL  SWTCH(1,2,3,4)                                                FET00162
     GO TO 10                                                            FET00163
C                                                                        FET00164
C                                                                        FET00165
 50  J = 1                                                               FET00166
     DO 52 I = 1,4                                                       FET00167
     IF (K(1) - K(J)) 51, 52, 52                                         FET00168
 51  J = I                                                               FET00169
 52  CONTINUE                                                            FET00170
     GO TO (90, 54, 55, 56),J                                            FET00171
 54  CALL  SWTCH(1,2,3,4)                                                FET00172
     GO TO 90                                                            FET00173
 55  CALL  SWTCH(1,3,2,4)                                                FET00174
     GO TO 90                                                            FET00175
 56  CALL SWTCH(1,3,2,4)                                                 FET00176
     GO TO 54                                                            FET00177
 90  IA = K(1)                                                           FET00178
     IC = K(3)                                                           FET00179
     IAB = K(1) + K(2)                                                   FET00180
     ICD = K(3) + K(4)                                                   FET00181
     RETURN                                                              FET00182
     END                                                                 FET00183
     SUBROUTINE  SWTCH(I,J,K,L)                                          FET00184
     COMMON /SWHCOM/ M(4)                                                FET00185
```

```
      IDUM = M(I)                                                       FET00186
      M(I) = M(J)                                                       FET00187
      M(J) = IDUM                                                       FET00188
      IDUM = M(K)                                                       FET00189
      M(K) = M(L)                                                       FET00190
      M(L) = IDUM                                                       FET00191
      RETURN                                                            FET00192
      END                                                               FET00193
      SUBROUTINE TABLE                                                  FET00194
C-----------------------------------------------------------------------FET00195
C     AUTHOR: DAVID L. TRITCHLER                                        FET00196
C     PURPOSE: OUTPUT CONTINGENCY TABLE WITH MARGINAL TOTALS            FET00197
C-----------------------------------------------------------------------FET00198
      DIMENSION IGRP(9), ICLAS(9) , INT(4)                              FET00199
      COMMON IA, IC, IAB, ICD, IAC, ICUT                                FET00200
      DATA     IGRP    / 2*1H , 1HG, 1HR, 1HO, 1HU, 1HP, 2*1H /         FET00201
      DATA     ICLAS   / 2*1H , 1HX, 3*1H , 1HY, 2*1H /                 FET00202
      EQUIVALENCE ( IA, INT(1) )                                        FET00203
    1 FORMAT ( 1H1, 23X, 7HOUTCOME / 19X, 1H+, 12X, 1H- )               FET00204
    2 FORMAT (8X,A1,4X, 2( 1H+, 12( 1H- ) ), 1H+ )                      FET00205
    3 FORMAT ( 8X, A1, 4X, 2( 1HI,12X), 1HI )                           FET00206
    4 FORMAT ( 8X, A1, 2X, A1, 3( 1X, 1HI, 1X, I10 ) )                  FET00207
    5 FORMAT ( 1H+, 39X, 11(1H-) / 39X, 1HI / 12X, 2(3X, I10 ),         FET00208
     C       1X, 1HI, 1X, I10 / 39X, 1HI )                              FET00209
      IBD = 0                                                           FET00210
      WRITE ( IOUT, 1 )                                                 FET00211
      K = 0                                                             FET00212
   20 K = K + 1                                                         FET00213
      GO TO ( 50, 60, 70, 60, 50, 60, 70, 60, 50, 30 ), K               FET00214
   30 N = IAC + IBD                                                     FET00215
      WRITE ( IOUT, 5 ) IAC, IBD, N                                     FET00216
      RETURN                                                            FET00217
   50 WRITE ( IOUT, 2 ) IGRP(K)                                         FET00218
      GO TO 20                                                          FET00219
   60 WRITE ( IOUT, 3 ) IGRP(K)                                         FET00220
      GO TO 20                                                          FET00221
   70 I = K - (K / 3) * 3 + 1                                           FET00222
      J = I + 2                                                         FET00223
      IDIF = INT(J) - INT(I)                                            FET00224
      WRITE ( IOUT, 4 ) IGRP(K), ICLAS(K), INT(I), IDIF, INT(J)         FET00225
      IBD = IBD + IDIF                                                  FET00226
      GO TO 20                                                          FET00227
      END                                                               FET00228
```

HASP-II OUTPUT COMPLETE FOR JCLSE5 , JOB 644

```
      RETURN                                                            FET00248
   50 WRITE ( IOUT, 2 ) IGRP(K)                                         FET00249
      GO TO 20                                                          FET00250
   60 WRITE ( IOUT, 3 ) IGRP(K)                                         FET00251
      GO TO 20                                                          FET00252
   70 I = K - (K / 3) * 3 + 1                                           FET00253
      J = I + 2                                                         FET00254
      IDIF = INT(J) - INT(I)                                            FET00255
      WRITE ( IOUT, 4 ) IGRP(K), ICLASS(K), INT(I), IDIF, INT(J)        FET00256
      IBD = IBD + IDIF                                                  FET00257
      GO TO 20                                                          FET00258
      END                                                               FET00259
C                                                                       FET00260
C                                                                       FET00261
C                                                                       FET00262
C                                                                       FET00263
C                                                                       FET00264
C  THE FOLLOWING SUBROUTINE SHOULD BE USED ONLY IN IBM 360/370 OS       FET00265
C     FORTRAN IV                                                        FET00266
C                                                                       FET00267
C                                                                       FET00268
      SUBROUTINE USERP(I1,I2,D,I)                                       FET00269
      COMMON IA, IC, IAB, ICD, ICO, ICUT, N80    , NNT(5)               FET00270
      COMMON KARD(200)                                                  FET00271
      COMMON IE                                                         FET00272
      REAL*8 D                                                          FET00273
      I1 = 0                                                            FET00274
      CALL TABLE                                                        FET00275
      PRINT 1                                                           FET00276
    1 FORMAT('0COMPUTATION EXCEEDS PROGRAM LIMITATIONS')                FET00277
      IE = 1                                                            FET00278
      RETURN                                                            FET00279
      END                                                               FET00280
```

8. SUBROUTINES

(a) CNVRT - converts character input into integers

(b) DRECT - puts problem into the standard format (see Mathematics Discussion).

(c) SWITCH- Call by subroutine DRECT to manipulate values.

(d) TABLE - prints a 2X2 contingency table of the input data along with the resultant probability value.

(e) USERR - This subroutine should be included only if the source program is compiled using IBM 360/370 OS FORTRAN IV. Its purpose is to capture errors when overflow occurs and print the message "COMPUTATION EXCEEDS PROGRAM LIMITATIONS" rather than allowing the operating system to take action. This subroutine may be omitted. Then normal system overflow action will be taken.

9. MEMORY REQUIREMENTS AND RUN TIME

This program is compiled on an IBM 360/65, and requires 30K to run. This could be reduced by

1) deleting subroutine CNVRT (which reads free-field input) and entering data in fixed format.

2) deleting sub-routine TABLE and merely listing the input data and the probability without printing a table.

This program requires less than 1 second of CPU time to execute a given problem on an IBM 360/65.

10. BIBLIOGRAPHY

1. W.H. Robertson, Programming Fisher's Exact Method of Comparing Two Percentages, Technometrics, vol.2, no.1, 1960, p.103.

2. S. Siegel, Nonparametric Statistics For the Behavioral Sciences, McGraw - Hill, New York, 1956.

3. H.M. Blalock , Social Statistics, McGraw - Hill, New York, 1972.